

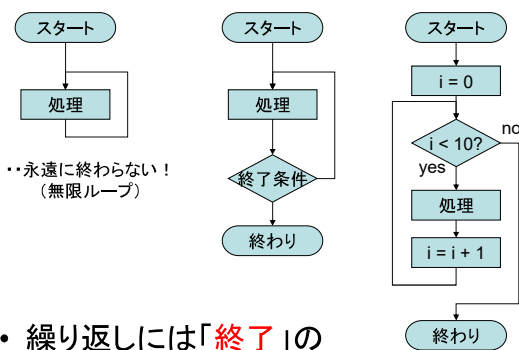
# 情報処理演習 (3)繰り返し

知能システム学 准教授  
万 偉偉(ワン ウエイウエイ)

## 繰り返しとは

- プログラムの**同じ部分を何度も実行**する
  - 同じことを何回も書かなくていい
  - プログラムが見やすくなる
  - 実行する回数を変更しやすい
- 繰り返しのポイントは
  - 何回**繰り返すか
  - 繰り返し回数を**どのように管理**するか  
→ ここでも変数を用いる

## 繰り返し



- 繰り返しには「**終了**」のメカニズムが必要

## 繰り返しの書き方

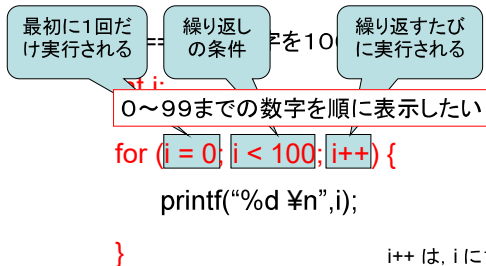
==== という文字を100行書きたい

```
int i;
for (i = 0; i < 100; i++) {
    printf("==== ¥n");
}
```

i++ は、iに1を足すことを表す  
(i = i + 1 と書いてもほぼ同じ)

- 最初 i に 0 を入れる
- i が 100 より小さい場合はブロックを実行
  - そうでない場合は下へ抜ける
- ブロックが終わると i に 1 を足す

## 繰り返しの書き方



i++ は、iに1を足すことを表す  
(i = i + 1 と書いてもほぼ同じ)

- 最初 i に 0 を入れる
- i が 100 より小さい場合はブロックを実行
  - そうでない場合は下へ抜ける
- ブロックが終わると i に 1 を足す

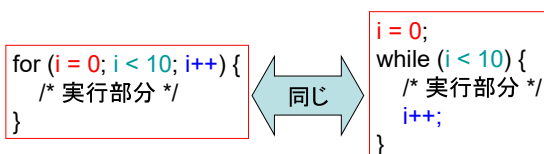
## 二重ループ

```
int i, j;
for (i = 0; i < 5; i++) {
    for (j = 0; j < 5; j++) {
        printf("i = %d, j = %d ¥n", i, j);
    }
}
```

- 繰り返しの繰り返し
  - 上の例では、printf は25回実行される。

## while

- while
  - for のうち、**条件判断部分**しかないもの



うまく使い分けるべし。

## do - while

- 条件判定を最後に行いたいときに使う
  - 処理の結果、やり直す場合など

```
do {
    printf("正の数を入力してください。 ¥n");
    scanf("%d", &a);
} while(a <= 0);
```

負の数か0が入力された場合、再度入力させる

## 繰り返しでよく使う省略形

```

i++  →  i = i + 1
++i  →  i = i + 1
i--  →  i = i - 1
--i  →  i = i - 1
i += 2 → i = i + 2
i -= 2 → i = i - 2
i *= 2 → i = i * 2
i /= 2 → i = i / 2
i %= 2 → i = i % 2
    
```

## i++と++iの違い

**i++** は i++ を含む式の処理が終了した後に i が1増。  
**++i** は ++i を含む式の処理の前に1増。  
 → i--, --i も同様。

```
int a, b;
int i=0;
```

```
a=i++;  → aは0, iは1になる.
b=++i;  → bは2, iは2になる.
```

## printf文. 高機能編

整数の場合:

書式: "%表示桁数d"  
 "表示桁数"のスペースに整数を右詰めで表示する。

```
例: printf("%4d\n", 40/3);
出力: 〇〇13
```

実数の場合:

書式: "%表示桁数.小数点以下の桁数f"  
 "表示桁数"のスペース(小数点"も含める)に、  
 小数点以下を"小数点以下の桁数"に丸めて  
 右詰めで表示する。

```
例: printf("%6.2f\n", 1.0/3.0);
出力: 〇〇0.33
```

## printf文. 高機能編

16進数で表記:

書式: "%x"  
 例: printf("%x\n", 15);  
 出力:

10進	2進	16進
0	0000	
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		

## printf文. 高機能編

16進数で表記:

書式: "%x"  
 例: printf("%x\n", 15);  
 出力:

10進	2進	16進
0	0000	
1	0001	
2	0010	
3	0011	
4	0100	
5	0101	
6	0110	
7	0111	
8	1000	
9	1001	
10	1010	
11	1011	
12	1100	
13	1101	
14	1110	
15	1111	

## printf文. 高機能編

16進数で表記:

書式: "%x"  
 例: printf("%x\n", 15);  
 出力:

10進	2進	16進
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	
11	1011	
12	1100	
13	1101	
14	1110	
15	1111	

## printf文. 高機能編

16進数で表記:

書式: "%x"  
 例: printf("%x\n", 15);  
 出力:

10進	2進	16進
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

## 今日学んだ内容

- 繰り返し
  - for(初期化, 繰り返し条件, 増減処理) { }
  - while(繰り返し条件) { }
  - do { } while(繰り返し条件)   ⇒ 必要な文字は?
- 二重ループ
 

```
for (i = 0; i < 3; i++)
    for (j = 0; j < 4; j++)
        printf("i = %d, j = %d \n", i, j);
```

 ⇒ 何回実行?

## 今日学んだ内容

- `i++` と `++i`

```
int a=0, b=0;
```

```
int i=0;
```

```
a=i++;
```

```
b=++i;
```

⇒ aとbの値は？

- 四則演算の省略形

```
a=a*8;
```

⇒ 省略形は？

- `printf` 高機能編

```
printf(" %f ",a);
```

⇒ 整数は5桁、小数第3位まで出力するには？